

```
1 <?php
2
3 /**
4  * @author GOLAY Brian
5  * @version 1.0 (2021/05/20)
6  * Flight edit logic here
7  */
8 require_once "../sql/userDAO.php";
9 require_once "../sql/flightDAO.php";
10 require_once "../sql/dbConnection.php";
11 require_once "../sql/mediaDAO.php";
12
13 use FlightClub\sql\FlightDAO;
14 use FlightClub\sql\DBConnection;
15 use FlightClub\sql\MediaDAO;
16 use FlightClub\sql\userDAO;
17
18 //we check if the user is already logged-in
19 if (!isset($_SESSION['userID'])) {
20     header("Location: ../index.php?page=homepage");
21     exit();
22 }
23
24 //we hide error to the user to only show our custom errors
25 error_reporting(0);
26
27 //filter the id from the url
28 $flightId = filter_input(INPUT_GET, "id", FILTER_SANITIZE_NUMBER_INT);
29 //we filter the user input
30 $role = filter_input(INPUT_POST, "Cd_Role", FILTER_SANITIZE_STRING);
31 $flight = filter_input(INPUT_POST, "No_Flight", FILTER_SANITIZE_STRING);
32 $dateDeparture = filter_input(INPUT_POST, "Dttm_Departure", FILTER_SANITIZE_STRING);
33 $dateArrival = filter_input(INPUT_POST, "Dttm_Arrival", FILTER_SANITIZE_STRING);
34 $timeEngineOn = filter_input(INPUT_POST, "Tm_Engine_On", FILTER_SANITIZE_STRING);
35 $timeEngineOff = filter_input(INPUT_POST, "Tm_Engine_Off", FILTER_SANITIZE_STRING);
36 $typeAircraft = filter_input(INPUT_POST, "Nm_Plane", FILTER_SANITIZE_STRING);
37 $registrationPlane = filter_input(INPUT_POST, "No_Plane", FILTER_SANITIZE_STRING);
38 $icaoDeparture = filter_input(INPUT_POST, "Cd_ICAO_Departure",
39     FILTER_SANITIZE_STRING);
40 $icaoArrival = filter_input(INPUT_POST, "Cd_ICAO_Arrival", FILTER_SANITIZE_STRING);
41 $flightType = filter_input(INPUT_POST, "Cd_Flight_Type", FILTER_SANITIZE_STRING);
42 $flightMode = filter_input(INPUT_POST, "Cd_Flight_Mode", FILTER_SANITIZE_STRING);
43 $weather = filter_input(INPUT_POST, "Txt_Meteo", FILTER_SANITIZE_STRING);
44 $passengers = filter_input(INPUT_POST, "Nb_Passengers", FILTER_SANITIZE_NUMBER_INT);
45 $notes = filter_input(INPUT_POST, "Txt_Note", FILTER_SANITIZE_STRING);
46 $btn = filter_input(INPUT_POST, 'btnModify', FILTER_SANITIZE_STRING);
47 $imgToDelete = filter_input(INPUT_POST, 'deletedImg', FILTER_SANITIZE_STRING);
48
49 $validTime = true;
50 $engineTime = true;
51
52 $goodDateTimeDepartureFormat = "";
53 $goodDateTimeArrivalFormat = "";
54 $resetTimeDeparture = "";
55 $resetTimeArrival = "";
56 $resetTimeEngineOn = "";
57 $resetTimeEngineOff = "";
58 $error = "";
59 /**
```

```
59  * Function found on Stack overflow to check if a time is valid. Found at:
    https://stackoverflow.com/questions/3964972/validate-that-input-is-in-this-time-
    format-hhmm
60  *
61  * @param string $time
62  * @return bool
63  */
64  function checkTime($time)
65  {
66      //check that the first char is either 0, a 1 or a 2, then the second character
    between 0 and 9. Then it looks for the ":" character. Then it check that the first
    minute number is between 0 and 5, finally, it check that the last number is between 0
    and 9
67      if (preg_match("/^(?:2[0-3]|[01][0-9]):[0-5][0-9]$/", $time)) {
68          return true;
69      } else {
70          return false;
71      }
72  }
73
74  /**
75   * function that validate the date
76   *
77   * @param string $date
78   * @return bool
79   */
80  function validateDate($date)
81  {
82      $newDate = explode("T", $date);
83      return checkdate(explode("-", $newDate[0])[1], explode("-", $newDate[0])[2],
    explode("-", $newDate[0])[0]);
84  }
85
86  //redirect the user if the id is wrong
87  if (empty($flightId)) {
88      header("Location: ../index.php?page=404");
89      exit();
90  }
91
92  //we initialize the variable that allow the user to see this page
93  $isAllowedToSeeThisFlight = false;
94
95  //we get the data of the pilot of this flight
96  $pilotOfThisFlight = userDao::getUserByFlightID($flightId);
97
98  //if the pilot is the current user, we allow him to see the page
99  if ($pilotOfThisFlight['Id_User'] == $_SESSION['userID']) {
100      $isAllowedToSeeThisFlight = true;
101  }
102
103  //if the user is not allowed to see the page, we show him a 404
104  if (!$isAllowedToSeeThisFlight) {
105      header("Location: ../index.php?page=404");
106      exit();
107  }
108
109  //we get all flight from the current user
110  $userFlight = FlightDAO::getFlightById($flightId);
111
112  //we reset the message
```

```
113 $message = array();
114 $successMessage = "";
115 //if all data are not empty
116 if (!empty($btn)) {
117     if (
118         !empty($role) && !empty($flight) && !empty($dateDeparture) &&
119         !empty($dateArrival) &&
120         !empty($typeAircraft) && !empty($registrationPlane) && !empty($icaoDeparture)
121         && !empty($icaoArrival) && !empty($flightType)
122         && !empty($flightMode) && !empty($weather) && $passengers != ""
123     ) {
124     } else {
125         $message[] = "Entrez tout les champs";
126     }
127     //we check that the data is valid
128     if (strlen($icaoDeparture) == 4 && strlen($icaoArrival) == 4) {
129     } else {
130         $message[] = "Vérifiez les codes ICAO";
131     }
132     //check that dates are valid
133     if (validateDate($dateDeparture) && validateDate($dateArrival)) {
134     } else {
135         $message[] = "La date n'est pas valide";
136     }
137     //check the validity of the hours of the take off time and landing time
138     if (checkTime(explode(":", (explode("T", $dateDeparture)[1]))[0] . ":" .
139     explode(":", (explode("T", $dateDeparture)[1]))[1]) && checkTime(explode(":",
140     (explode("T", $dateArrival)[1]))[0] . ":" . explode(":", (explode("T", $dateArrival)
141     [1]))[1])) {
142         $validTime = true;
143         //check the validity of the hours for the engine off and engine on time
144         if (!empty($timeEngineOn) && !empty($timeEngineOff)) {
145             $engineTime = true;
146             if (checkTime(explode(":", $timeEngineOn)[0] . ":" . explode(":",
147             $timeEngineOn)[1]) && checkTime(explode(":", $timeEngineOff)[0] . ":" . explode(":",
148             $timeEngineOff)[1])) {
149                 $validTime = true;
150             } else {
151                 $validTime = false;
152             }
153         } else {
154             $timeEngineOff = null;
155             $timeEngineOn = null;
156         }
157     } else {
158         $message[] = "La date et l'heure saisie ne sont pas valides";
159     }
160     if ($validTime && $engineTime) {
161         //check that the type of aircraft doesn't blow the database
162     } else {
163         $message[] = "L'heure saisie n'est pas valide";
164     }
165     if (strlen($typeAircraft) <= 45) {
166         //check that the registration of the aircraft doesn't blow the database
167     } else {
168         $message[] = "Le type d'avion est trop long";
169     }
170     if (strlen($registrationPlane) <= 45) {
171         //check that the flight type doesn't blow the database
172     } else {
```

```
166     $message[] = "L'immatriculation de l'avion est trop longue";
167 }
168 if (strlen($flightType) <= 45) {
169     //check that the flight mode doesn't blow the database
170 } else {
171     $message[] = "Le type de vol est trop long";
172 }
173 if (strlen($flightMode) <= 45) {
174     //check that the weather doesn't blow the database
175 } else {
176     $message[] = "Le mode de vol est trop long";
177 }
178 if (strlen($weather) <= 512) {
179     //check that the passengers number doesn't blow the database
180 } else {
181     $message[] = "La météo est trop longue";
182 }
183 if ($passengers <= 9999999999 && $passengers >= 0) {
184     //check that the notes doesn't blow the database
185 } else {
186     $message[] = "Le nombre de passagers est invalide";
187 }
188 if (strlen($notes) <= 1024) {
189     //every data is correct
190 } else {
191     $message[] = "Les notes sont trop longues";
192 }
193
194 //we check that every data is ok
195 if (empty($message)) {
196
197     //we check with is the latest id in the database
198     $allflight = FlightDAO::getAllFlightOrderById();
199     try {
200
201         //converting departure datetime to a good datetime
202         $dateDeparture = explode("T", $dateDeparture);
203         $dateDeparture = $dateDeparture[0] . " " . $dateDeparture[1];
204
205         //converting arrival datetime to a good datetime
206         $dateArrival = explode("T", $dateArrival);
207         $dateArrival = $dateArrival[0] . " " . $dateArrival[1];
208
209         //we edit the flight
210         FlightDAO::editFlight(
211             $flightId,
212             $_SESSION['userID'],
213             $role,
214             strtoupper($flight),
215             $dateDeparture,
216             $dateArrival,
217             $timeEngineOn,
218             $timeEngineOff,
219             $typeAircraft,
220             strtoupper($registrationPlane),
221             strtoupper($icaoDeparture),
222             strtoupper($icaoArrival),
223             $flightType,
224             $flightMode,
225             $weather,
```

```

226         $passengers,
227         $notes
228     );
229     //we show a success message
230     $successMessage = "Vol modifié avec succès!";
231
232     //we get the new data to show to the user
233     $userFlight = FlightDAO::getFlightById($flightId);
234 } catch (\Throwable $th) { //we show an error message
235     $message = "Une erreur est survenue. merci de réessayer";
236 }
237
238 //we check if there is some images to delete
239 if (!empty($imgToDelete)) {
240
241     $isAllowedToDeletePicture = false;
242
243
244
245
246     DBConnection::startTransaction();
247     $imgToDelete = explode(",", $imgToDelete);
248     unset($imgToDelete[count($imgToDelete) - 1]);
249     $pictureData = "";
250     // we try to remove each image
251     try {
252         foreach ($imgToDelete as $key => $value) {
253             $pictureData = MediaDAO::read_media_by_id($value);
254
255             if (userDAO::getUserByFlightID($pictureData['Id_Flight'])
256 ['Id_User'] == $_SESSION['userID']) {
257                 $isAllowedToDeletePicture = true;
258             }else {
259                 $isAllowedToDeletePicture = false;
260             }
261
262             if ($isAllowedToDeletePicture) {
263                 MediaDAO::del_mediaByIdMedia($value);
264                 unlink($pictureData['Txt_File_Path']);
265             }else {
266                 $message[] = "Vous n'avez pas la permission de supprimer
267 l'image!";
268             }
269
270             }
271             DBConnection::commit();
272         } catch (\Throwable $th) { //if there was an error, we rollback
273             DBConnection::rollback();
274             $message[] = "Une erreur est survenue lors de la suppression du
275 media";
276         }
277     }
278
279     //FILE UPLOAD
280
281     $name = "";
282     $lienimg = "";
283     $default_dir = "../media/";

```

```

283     $errorimg = "";
284     $size_total = 0;
285     $error = "";
286     $type = '';
287     $extension = '';
288     $nb_files = count($_FILES['media']['name']);
289     $MAX_FILE_SIZE = 3145728; // 3MB in bytes
290     $MAX_POST_SIZE = 73400320; // 70MB in bytes
291
292     $extensions = array(
293         "image" => array('.png', '.gif', '.jpg', '.jpeg'),
294     );
295     // available types of file
296     $types = array('image');
297
298     // we check that the user have clicked on the button
299     if ($btn != 'null') {
300         if ($_FILES['media']['name'][0] != "") {
301             DBConnection::startTransaction();
302             //we check the size of the image
303             foreach ($_FILES['media']['size'] as $key => $value) {
304                 if ($value > $MAX_FILE_SIZE) {
305                     $error = 'File too heavy.';
306                     DBConnection::rollback();
307                 } else {
308                     $size_total += $value;
309                 }
310             }
311             //if there is images we check each media size
312             if (isset($_FILES['media'])) {
313                 for ($i = 0; $i < $nb_files; $i++) {
314                     $errorimg = $_FILES['media']["error"][$i];
315                     if ($error == 'File too heavy.' || $size_total >
316 $MAX_POST_SIZE) {
317                         $error = "Fichier trop volumineux!";
318                         DBConnection::rollback();
319                     } else {
320                         //we check the mime type
321                         $separator = '/';
322                         $extension = explode($separator,
323 mime_content_type($_FILES['media']['tmp_name'][$i]))[1];
324                         $type = explode($separator,
325 mime_content_type($_FILES['media']['tmp_name'][$i]))[0];
326
327                         if (!in_array($type, $types)) {
328                             $error = "erreur dans le type de fichier";
329                             DBConnection::rollback();
330                         } else {
331                             if ($error != "erreur dans le type de fichier") {
332                                 if ($errorimg[0] == 0) { //if there is no error,
333 we add the file
334                                     $tmp_name = $_FILES['media']["name"][$i];
335                                     $name = explode(".", $tmp_name);
336                                     $name = $name[0] . uniqid() . "." . $name[1];
337
338                                     if (move_uploaded_file($_FILES['media']
339 ["tmp_name"][$i], $default_dir . $type . "/" . $name)) {
340                                         //add the filename in the database
341                                         MediaDAO::changePath($name,
342 $tmp_name[$i]);

```

```

337     $lienimg = $default_dir . $type . "/" .
$name;
338
339     } else {
340         DBConnection::rollback();
341     }
342     try { //we add the media in the database
343         MediaDAO::addmedia($lienimg, $flightId);
344
345         DBConnection::commit();
346     } catch (\Throwable $th) {
347         $error = $th;
348         DBConnection::rollback();
349     }
350     } else {
351         DBConnection::rollback();
352     }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361
362 if ($error != "") {
363     $message .= " " . $error;
364 }
365
366
367
368 //we get the new data of the pilot of this flight
369 $pilotOfThisFlight = userDAO::getUserByFlightID($flightId);
370
371 /**
372  * Function that compute a flight time on with date and hour. Found logic on
373  * stackoverflow and adapted it a bit:
374  * https://stackoverflow.com/questions/5463549/subtract-time-in-php
375  *
376  * @param string $Dt_Departure
377  * @param string $Dt_Arrival
378  * @return void
379  */
380 function computeTotalTime($Dt_Departure, $Dt_Arrival)
381 {
382     $start = strtotime($Dt_Departure);
383     $end = strtotime($Dt_Arrival);
384
385     //If you want it in minutes, you can divide the difference by 60 instead
386     $mins = (int)(( $end - $start) / 60);
387     echo $mins . ' minute(s) ' . '<br>';
388 }
389
390 /**
391  * Function that get all pictures from a flight with the id of the flight
392  *
393  * @param int $idFlight
394  * @return void
395  */

```

```
394 function showAllPicturesFromTheFlight($idFlight)
395 {
396     $picture = MediaDAO::readMediaByIdFlight($idFlight);
397     foreach ($picture as $key => $value) {
398         echo "<tr id='" . $value['Id_Picture'] . "'><td><img class='img-thumbnail'
src='" . $value['Txt_File_Path'] . "'></td><td><button type='button' class='btn btn-
danger' name='delete' value='" . $value["Id_Picture"] . "' onclick='deleteJS(" .
$value["Id_Picture"] . ", " . $idFlight . ")'>X</button></td></td></tr>";
399     }
400     if (empty($picture)) {
401         echo " <tr><td>Aucune image pour ce vol</td></tr>";
402     }
403 }
404
405 //change the format of the datetime to be a valid sticky datetime format
406 $goodDateTimeDepartureFormat = explode(" ", $userFlight['Dttm_Departure']);
407 $goodDateTimeDepartureFormat = $goodDateTimeDepartureFormat[0] . "T" .
$goodDateTimeDepartureFormat[1];
408
409 $goodDateTimeArrivalFormat = explode(" ", $userFlight['Dttm_Arrival']);
410 $goodDateTimeArrivalFormat = $goodDateTimeArrivalFormat[0] . "T" .
$goodDateTimeArrivalFormat[1];
411
```